VLSI Implementation of RSA Encryption System Using Ancient Indian Vedic Mathematics

Soumya Sadanandan
M.Tech VLSI & Embedded Systems
Mangalam College of Eng.
Kottayam
soumya.swaralayam@gmail.com

Ms. Anjali.V
Asst.Professor
Mangalam College of Eng.
Kottayam

ABSTRACT
The standard techniques for providing privacy & security in data networks include encryption/decryption algorithms such as Advanced Encryption System(AES) (private key) & RSA (public key).RSA is one of the safest standard algorithms,based on public key,for providing security in networks.One of the most time consuming process in RSA encryption/decryption algorithm is the computation of a^b mod n where a is the text,(b,n) is the key & this paper examines how this computation could be speeded up drawing up on the Indian Vedic Mathematics.

General Terms
RSA encryption/decryption, Vedic Mathematics

1. INTRODUCTION
The standard techniques for providing privacy and security in data networks include encryption/decryption algorithms such as Advanced Encryption System (AES) (private-key) and RSA (public-key). RSA is one of the safest standard algorithms, based on public-key, for providing security in networks. While hardware implementation of this algorithm tends to be faster compared to its software counterpart, there is a scope for further improvement of performance of RSA hardware. One of the most time consuming processes in RSA encryption/decryption algorithm is the computation of a^b mod n where a is the text , (b,n) is the key and this paper examines how this computation could be speeded up drawing up on the Indian Vedic Mathematics.

2. FUNDAMENTAL MATH OF RSA

2.1 Public Key Cryptography

From the original RSA paper by R.L. Rivest, A. Shamir,& L. Adleman , an asymmetric cryptographic system was proposed that uses modular exponentiation for encryption and decryption. For ease of implementation it was proposed that both the encryption and decryption functions be identical; the only difference being the inputdata. The term 'asymmetric cryptographic' means that the keys used for encryption and decryption are different.In the case of RSA, the encryption key is made publicly available and the decryption key is kept private.The encryption/decryption chain is described as followswhere M is the original message data, C is the encrypted message or cipher text, e, m is the publicly available encryption key, and f is the decryption key:

$$C = M_e \bmod m$$

$$M = C_f \bmod m$$

Key generation for RSA starts with the selection of two prime numbers which are then multiplied together to produce the publicly visible modulus m:

$$m = p \oplus q$$

The strength of RSA is based on the difficulty of factoring m to discover the original primes p,q. Hence the larger the value of these primes, the harder the factorisation problem becomes. Again, typical values for these primes are 512 to 4096bits with the later exponentially stronger than the former.Next an integer, e, that is relatively prime to $(p\text{-}1)(q\text{-}1)$, ischosen as the public key. Typically and for practical reasons, e can be one of the first Fermat Numbers 3, 5, 17, 33 which will always satisfy the following:

$mx=(p\text{-}1)\ (q\text{-}1)$

$$\gcd(e, mx) = 1$$

gcd=greatest common divisor

To generate the private key, f, it is then necessary to find the multiplicative inverse of $e \bmod mx$.

$$(f.e) \bmod mx = 1$$

$$f = Euc\ (e, mx)$$

Euc =Euclid's Algorithm

Finally, publish e, m as the public key and keep f secret. Ideally, the values of p, q, the generator primes should be destroyed.

2.2 RSA ALGORITHM

RSA algorithm can generally be further classified into key generation algorithm, encryption algorithm, and decryption algorithm.The RSA key generation algorithm can be described in the following five steps.

1. Generate two large random and distinct primes P and Q
2. Calculate N = P.Q and K= (P – 1)(Q– 1)
3. Choose a random integer J, 1 < J <K, such that gcd(J,K) = 1
4. Compute the unique integer I, 1 < I < K, such that JI=1 (mod K)
5. Public key is (N, J) and private key is (N, I)

While RSA encryption algorithm is described by
L = MJ mod N,
The decryption algorithm is described by
M = LI mod N,
where L represents the cipher text and M represents the message

2.2 MULTIPLER ARCHITECTURE

The Multiplier Architecture is based on the Vertical and Crosswise algorithm of ancient Indian Vedic Mathematics. In the overlay architecture, grouping of 4 bits at a time is done for both multiplier and multiplicand and thereafter vertical and crosswise algorithm is applied to decompose the whole of the multiplication operation into 4x4 multiply modules. The algorithm is explained in Table-1 for 16x16 bit number. After getting the sub-product bits in parallel from the 4x4 multiply modules, we can employ an efficient method of addition to generate the final 32 bit product. This method can be generalized for NXN bit multiplication where N is a multiple of 4 such as 8,12,16,20,24,....4n. Thus instead of implementing the entire multiplication through a single NXN bit multiplier, we can get the same product efficiently by using the proposed overlay architecture. The advantage of this is that the multiply operation of large number of bits can now be performed by using smaller and efficient 4x4 multiplier.

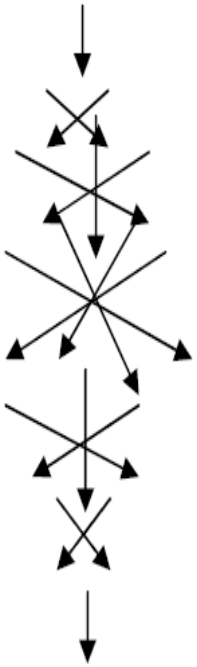
TABLE 1- 8 x 8 bit Vedic multiplier Using Urdhva Tiryakbhyam
CP- Cross Product (Vertically and Crosswise)

A= A7 A6 A5 A4 A3 A2 A1 A0
X1 X0
B= B7 B6 B5 B4 B3 B2 B1 B0
Y1 Y0

X1 X0 Multiplicand [8 bits]
Y1 Y0 Multiplier [8 bits]

F E D C
P3 P2 P1 P0 Product [16 bits]

Where X1, X0,Y1 and Y0 are each of 4 bits.



Note: Each Multiplication operation is an embedded parallel 4x4 multiply module

3. SIMULATION RESULTS

The RSA encryption/decryption circuitry achieves a significant improvement in performance using the Vedic hierarchical multiplier. The RSA circuitry has less timing delay compared to its implementation using traditional multipliers.It has already been demonstrated that Vedic hierarchical multiplier is efficient than conventional multipliers in terms of area/speed . For the Xilinx Virtex 4 family, it is found that the gate delay for RSA circuitry using 8x8 vedic multiplier architecture is 20.90ns.While the gate delay of RSA circuitry using 8x8 booth multiplier is 48.480ns.

4. CONCLUSION

The RSA circuitry implemented with Vedic high speed multiplier exhibits improved efficiency in terms of speed and area. Due to its parallel and regular structure the proposed architecture can be easily laid out on silicon chip. It has the advantage that as the number of bits increases its gate delay and area increase very slowly as compared to RSA circuitry employing traditional multipliers. It is found that this design is quite efficient in terms of silicon area and speed and should result in substantial savings of resources in hardware when used for crypto and security applications. The combinational delay obtained for RSA circuitry using vedic multiplier is much lesser than by using booth multiplier. The results suggest that Vedic multiplier is an extreme fast multiplier & is well ahead of booth multiplier.Concluded that RSA circuitry implemented using vedic multiplier proves to be highly efficient in terms of speed & area.

5. ACKNOWLEDGEMENT

My sincere thanks to Peiyi Zhao,Member IEEE

6. REFERENCE

[1] Himanshu Thapliyal and M.B Srinivas,VLSI Implementation of RSA Encryption System Using Ancient Indian Vedic Mathematics,Center for VLSI and Embedded System Technologies,International Institute of Information Technology

[2] H Thapliyal,and H R Arabnia,A time area-power efficient multiplier and square architecture based on Ancient Indian Vedic Mathematics, Proceedings of the 2004 International Conference on VLSI,June 2004,pp. 434-9.

[3] M C Hanumantharaju,H Jayalaxmi,R K Renuka,and M Ravishankar,A high speed block convolution using Anciet Indian Vedic Mathematics,International Conference on Computational Intelligence and Multimedia.